# Introduction

The aim of this study is to predict the impact of climate change in Egypt using machine learning methods.

Machine learning is a type of artificial intelligence that allows computers to learn from data and make predictions or decisions without being explicitly programmed to do so and it has been widely used in various fields, including climate science.[1]

# Data

We obtained data from the World Bank Climate Change Knowledge Portal.[2] Before conducting our analysis, we performed some basic preprocessing on the data. Data preprocessing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we preprocess our data before feeding it into our model.[3] Although the dataset was already in good shape due to its approved source, we still filled in any missing values by calculating the minimum, maximum, and mean values for each variable and using these values to fill in any missing data. Additionally, we removed any observations that were completely empty.

Our dataset included independent variables such as **minimum temperature, maximum temperature, mean temperature, CO2 levels,** and **precipitation**. These factors were used to predict the impact of climate change. Our target variables included **sea level (measured in millimeters), wheat yield (measured in metric tons), rice yield (measured in metric tons), cotton yield (measured in metric tons),** and **mortality under 5 years (measured as the number of deaths per 1000 births)** these outcomes were the focus of our predictions.

# Methods

In this study, we used several machine learning methods from the scikit-learn library to predict the impact of climate change in Egypt. These methods are algorithms that can be learned from data and make predictions. We used a variety of regression models to evaluate our predictions and understand the relationship between our target variables and independent variables.

scikit-learn is a popular open-source library for machine learning in Python [4]. It provides a wide range of tools for data analysis and modeling, including classification, regression, clustering, and dimensionality reduction. scikit-learn is built on top of other widely used libraries such as NumPy, SciPy, and matplotlib. Our focus is on regression models because our dataset contains continuous variables (numerical values).

Furthermore, scikit-learn provides a consistent interface for its machine learning algorithms. This makes it easy to switch between different algorithms and compare their performance. scikit-learn also provides tools for model selection and evaluation, data preprocessing, and feature extraction.

One of the methods we used is called LASSO (Least Absolute Shrinkage and Selection Operator) [5]. LASSO is a method that can be used to predict a numerical value (such as sea level) based on several input variables (such as temperature, CO2 and Precipitation).

We also used other machine learning methods in our analysis, such as Linear Regression [6], Ridge Regression [7], LassoLars [8], Random Forest Regression [9], and Tweedie Regressor [10]. These methods were chosen based on their ability to handle the specific characteristics of our dataset and the nature of the problem we are trying to solve.

Because we were dealing with multiple outputs (responses), we used the MultiOutputRegressor from scikit-learn [11] for better modeling. MultiOutputRegressor is a meta-estimator that extends single-output regression methods to multi-output problems. It works by fitting one regressor per target variable and making predictions for each target variable independently.

Linear Regression is a simple model that assumes a linear relationship between the independent and dependent variables. Ridge Regression is similar but adds a regularization term to prevent overfitting. Lasso Regression is another type of regularized Linear Regression that uses a different type of regularization term.

We also used other machine learning methods such as LassoLars, Random Forest Regression, and Tweedie Regressor. These methods were chosen based on their ability to handle the specific characteristics of our dataset and the nature of the problem we are trying to solve. Additionally, because we had multiple target variables, we used a Multioutput Regressor to make predictions for multiple target variables simultaneously. The MultiOutputRegressor from scikit-learn is a meta-estimator that extends single-output regression methods to multi-output problems by fitting one regressor per target variable and making predictions for each target variable independently.

LASSO (Least Absolute Shrinkage and Selection Operator) is a method that can be used to predict a numerical value based on several input variables. It works by finding the relationship between the input variables and the value we want to predict. This relationship is represented by a mathematical equation that can be used to make predictions.

In more detail, LASSO works by fitting a linear regression model to the data, but with an added constraint that the sum of the absolute values of the coefficients must be less than a certain value. This constraint has the effect of shrinking some coefficients towards 0, effectively excluding some input variables from the model. The strength of this constraint is controlled by a tuning parameter called λ. When λ is 0, LASSO is equivalent to ordinary least squares regression. As λ increases, more coefficients are shrunk towards 0.

The advantage of using LASSO is that it can automatically select which input variables are most important for making predictions. This can help to simplify the model and improve its interpretability.

# Model Evaluation

The performance of our models was evaluated using appropriate evaluation metrics for regression problems, such as mean squared error or mean absolute error. These metrics measure the difference between the predicted values and the observed values. The results show that LASSO regression performed the best among all the methods we tested.

We found that Lasso Regression worked particularly well for our dataset. This model uses a regularization term that encourages the coefficients of less important variables to be set to zero. This can help to prevent overfitting and improve the interpretability of the model.

We also found that some models worked better for some target variables than others. For example, one model might have made accurate predictions for sea level but less accurate predictions for wheat yield.

To evaluate the accuracy of our models, we used several different metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2) and Root Mean Squared Error (RMSE).

MSE measures the average squared difference between the predicted and actual values. It is calculated as follows: MSE = 1/n \* Σ(yi - ŷi)^2 where n is the number of observations, yi is the actual value and ŷi is the predicted value.

MAE measures the average absolute difference between the predicted and actual values. It is calculated as follows: MAE = 1/n \* Σ|yi - ŷi| where n is the number of observations, yi is the actual value and ŷi is the predicted value.

R-squared measures how well the model fits the data. It is calculated as follows: R2 = 1 - SSres/SStot where SSres is the sum of squared residuals and SStot is the total sum of squares.

RMSE measures the square root of the average squared difference between the predicted and actual values. It is calculated as follows: RMSE = sqrt(MSE).

We used these metrics to evaluate how well our models were able to make accurate predictions for each target variable.

* **MAE (Mean Absolute Error)**: This is the average of the absolute differences between the predicted and actual values. It measures the average magnitude of the errors in a set of predictions, without considering their direction. A lower MAE value indicates better model performance.
* **MSE (Mean Squared Error)**: This is the average of the squared differences between the predicted and actual values. It measures the average squared magnitude of the errors in a set of predictions. A lower MSE value indicates better model performance.
* **RMSE (Root Mean Squared Error)**: This is the square root of the MSE. It measures the standard deviation of the residuals (prediction errors). A lower RMSE value indicates better model performance.
* **R^2 (Coefficient of Determination)**: This is a statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s). It provides a measure of how well observed outcomes are replicated by the model. An R^2 value of 1 indicates that the model perfectly fits the data, while an R^2 value of 0 indicates that the model does not explain any of the variance in the data.

# Technologies

We used the Python programming language and the scikit-learn library to implement our models. We also used Jupyter Notebook as our programming environment. In addition to scikit-learn, we used several other libraries including numpy for numerical computing, pandas for data manipulation and analysis, matplotlib for data visualization and seaborn for statistical data visualization.

By using machine learning and regression analysis together, we were able to develop a robust methodology for predicting the impacts of climate change on Egypt. This approach allowed us to make accurate predictions and to understand the underlying factors that are driving these changes.

# Conclusion

In conclusion, our study demonstrates the potential of machine learning methods for predicting the impact of climate change in Egypt on sea level, crop yields, and child mortality. Further research is needed to refine and improve our models and to explore the use of additional data sources and machine learning methods.

In simpler terms, we used data on temperature, CO2 levels, and precipitation to make predictions about how climate change will affect sea levels, crop yields, and child mortality in Egypt. We then used statistical methods to evaluate the accuracy of these predictions.

# References

[1] <https://en.wikipedia.org/wiki/Machine_learning>, <https://www.ipcc.ch/report/ar5/wg1/>

[2] [add the link]

[3] [Introduction to Data Preprocessing in Machine Learning | by Dhairya Kumar | Towards Data Science](https://towardsdatascience.com/introduction-to-data-preprocessing-in-machine-learning-a9fa83a5dc9d)

[4] <https://scikit-learn.org/stable/>